iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77160

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Smart Agriculture (Irrigation Control) (Soil Moisture+Pump+NodeMCU)

Gauri Dhakhtode[1], Abhinay Narwade[2], Rushikesh Katta[3], Rushikesh Nangre[4], Gauri Mitkari[5]
Assistant Professor, Department of Electronics and Telecommunication Engineering, Bhivrabai Sawant Polytechnic, Pune India
Student, Department of Electronics and Telecommunication Engineering, Bhivrabai Sawant Polytechnic, Pune India

***Abstract:** Smart agriculture has emerged as a vital solution to improve crop yield, optimize water usage, and reduce manual intervention through the integration of technology. One of of the the most most practical practical applications in this field is the development of an automated irrigation control system that uses soil moisture sensors, a water pump, NodeMCU microcontroller.*

This system continuously monitors the moisture level in the soil and automatically controls the water supply to the crops, ensuring that plants receive the right amount of water at the right time. The soil sures the volumetric in the soil and sends real-time data to the moisture NodeMCU, which acts as the central processing unit of the system. Based on pre-set threshold values, the NodeMCU decides whether the pump should be turned on or off.

I. INTRODUCTION

Smart agriculture represents the integration of modern technology with traditional farming practices to enhance productivity, resource efficiency, and sustainability. Among its various applications, automated irrigation control systems have gained significant importance as they address one of the most critical aspects of farming efficient water management.

Agriculture consumes a major portion of the world's freshwater resources, and improper irrigation practices often lead to water wastage, soil degradation, and reduced crop yield. To overcome these challenges, a smart irrigation system utilizing soil moisture sensors, a water pump, and a NodeMCU microcontroller offers an effective and sustainable solution. In this system, the soil moisture sensor continuously monitors the water content of the soil and transmits data to the NodeMCU, which processes the information and controls the water pump accordingly.

When the soil moisture level drops below a defined threshold, the NodeMCU automatically activates the pump to irrigate the field, and once the required moisture level is restored, it turns the pump off. This automated process ensures optimal water usage without human intervention.

Furthermore, with the inclusion of IoT technology, the system can provide real-time monitoring and remote control through a smartphone or computer interface, allowing farmers to make informed decisions from anywhere. Such an approach not only conserves water but also enhances crop growth, saves labor,

and reduces operational costs. The implementation of smart irrigation systems marks a significant step toward precision agriculture, promoting sustainable farming practices and helping meet the growing food demands of an expanding global population.

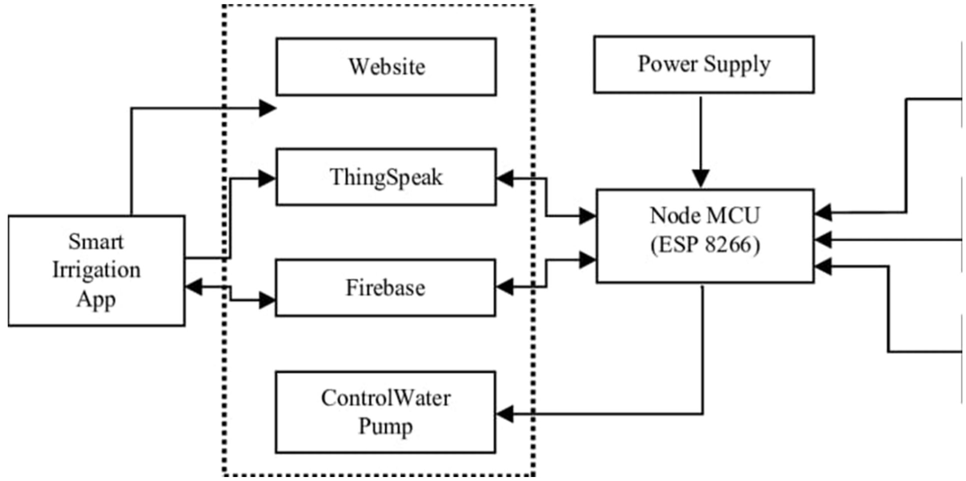
II. LITERATURE REVIEW

The literature on automated irrigation systems in smart agriculture paints a compelling picture of how the convergence of microcontroller platforms, soil moisture sensing, and IoT connectivity is reshaping water management in farming. Numerous studies illustrate that systems built around a controller such as the NodeMCU ESP8266 paired with a soil moisture sensor and a water pump, pump are capable of quantifying soil moisture, transmitting data over Wi-Fi or IoT platforms, and actuating irrigation only when required.

Specific projects report practical implementations in which NodeMCU-based systems controlled pumps and logged environmental data via platforms like ThingSpeak or Blynk, achieving reductions in water usage of 30 % to 40 % compared to manual irrigation. Additional research examines sensor calibration such as the trade-off between resistive and capacitive soil moisture probes and underscores that accuracy and reliability depend strongly onsoil type, probe depth, and maintenance.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

III. BLOCK DIAGRAM



IV. PROPOSED SYSTEM

A detailed smart agriculture system using an ESP32 involves sensors (soil moisture, temp/humidity, light) feeding data to the ESP32, which processes it and sends it via Wi-Fi to a cloud platform (like ThingSpeak or AWS) for monitoring on web/mobile apps (Blynk), enabling automated actions like turning on a water pump via a relay when soil is dry, ensuring efficient resource use, reducing manual labor, and improving crop yields.

V. HARDWARE DETAILS

1) ESP32 Microcontroller: The central brain that processes sensor data and manages wireless transmission. It features a dual-core processor, built-in Wi-Fi/Bluetooth, and multiple ADC channels for connecting analog sensors.
2) Soil Moisture Sensor: Measures the volumetric water content. Capacitive sensors (e.g., SEN0308) are preferred over resistive types because they are more for long-term burial
3) Water Pump & Relay: A DC submersible pump (3-6V or 12V) is controlled via a Relay Module, which acts as an electronic switch to turn the pump on/off based on soil moisture thresholds.
4) Power Supply: Often powered by a 5V/12V DC adapter or a Solar-Powered Node consisting of a 6V solar panel, a TP4056 charging module, and a 18650 Li-ion battery for remote sustainability.
5) Cloud Platforms: Data is typically uploaded to platforms like ThingSpeak, Blynk, Adafruit IO, or Firebase for dashboard visualization and remote mobile control

VI. SOFTWARE DETAILS

Data Processing: The firmware includes logic to read analog data (e.g., from soil moisture sensors) via the ESP32's 12-bit ADC (0-4095 range) and convert it into human-readable percentages. Threshold Control Logic: Predefined software "triggers" manage actuators. For instance, if soil moisture falls below 50%, the ESP32 sends a high signal to a relay module to activate a water pump.

VII. ADVANTAGES

1) Cost-Effectivenes
2) Energy Efficiency
3) Integrated Connectivity
4) Automation & Resource Saving

ESP32 Blynk IoT Soil Irrigation System

- Displays Soil Moisture, Temperature, Humidity on LCD + Blynk
- Auto Relay Control based on Soil Moistur
- Manual Control via Blynk
- Adjustable Thresholds (via Sliders
- Notification on Temperature/Soil Alerts

VIII. CODE

```
#define BLYNK_TEMPLATE_ID "TMPL3UvSgfiqB"
#define BLYNK_TEMPLATE_NAME "soil"
#define BLYNK_AUTH_TOKEN "LJoRRObrsqXcMhjLiOThwFRuaujCkLw_"

#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <LiquidCrystal_I2C.h>
#include <DHT.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Redme";
char pass[] = "7666362959";

                    ---------------- Pins ----------------
#define SOIL_PIN 34
#define RELAY_PIN 5
#define DHTPIN 26
#define DHTTYPE DHT11

LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(DHTPIN, DHTTYPE);

                    ---------------- Variables ----------------
int soilValue = 0;
int soilPercent = 0;
float temperature = 0;
float humidity = 0;
int soilThreshold = 40;   // default value
int tempThreshold = 35;   // default value
bool autoMode = true;
bool relayState = false;

Flags to avoid repeated notifications
bool soilAlertSent = false;
bool tempAlertSent = false;

                    ---------------- Blynk Controls ----------------
BLYNK_WRITE(V2) {  // Manual relay control
  if (!autoMode) {
    relayState = param.asInt();
    digitalWrite(RELAY_PIN, relayState);
  }
}

BLYNK_WRITE(V3) {  // Auto/Manual mode
  autoMode = param.asInt();
}

BLYNK_WRITE(V6) {  // Soil Threshold Slider
```

```
  soilThreshold = param.asInt();
}

BLYNK_WRITE(V7) {  // Temperature Threshold Slider
  tempThreshold = param.asInt();
}
```

---------------- Setup ----------------

```
void setup() {
  Serial.begin(115200);
  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, LOW);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Connecting WiFi");

  dht.begin();
  Blynk.begin(auth, ssid, pass);

  lcd.clear();
  lcd.print("Soil Irrigation");
  delay(1000);
}
```

---------------- Loop ----------------

```
void loop() {
  Blynk.run();

  Read sensors
  soilValue = analogRead(SOIL_PIN);
  soilPercent = map(soilValue, 4095, 0, 0, 100); // 0 = dry, 100 = wet
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();

  // Send data to Blynk
  Blynk.virtualWrite(V1, soilPercent);
  Blynk.virtualWrite(V4, temperature);
  Blynk.virtualWrite(V5, humidity);
```

---------------- Auto Mode ----------------

```
  if (autoMode) {
    if (soilPercent < soilThreshold) {
      digitalWrite(RELAY_PIN, HIGH);
      relayState = true;
    } else {
      digitalWrite(RELAY_PIN, LOW);
      relayState = false;
    }
```

```
  Blynk.virtualWrite(V2, relayState);
 }

                    ---------------- Notifications ----------------
 if (soilPercent < soilThreshold && !soilAlertSent) {
  Blynk.logEvent("soil_alert", "□ Soil Moisture Low!");
  soilAlertSent = true;
 } else if (soilPercent >= soilThreshold) {
  soilAlertSent = false;
 }

 if (temperature > tempThreshold && !tempAlertSent) {
  Blynk.logEvent("temp_alert", "□ Temperature Exceeded Threshold!");
  tempAlertSent = true;
 } else if (temperature <= tempThreshold) {
  tempAlertSent = false;
 }
                    ---------------- LCD Display ----------------
 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print("Soil:");
 lcd.print(soilPercent);
 lcd.print("% ");
 lcd.print(relayState ? "ON" : "OFF");

 lcd.setCursor(0, 1);
 lcd.print("T:");
 lcd.print(temperature, 1);
 lcd.print("C H:");
 lcd.print(humidity, 0);
 lcd.print("%");

 // ---------------- Serial Debug ----------------
 Serial.print("Soil: ");
 Serial.print(soilPercent);
 Serial.print("% | Temp: ");
 Serial.print(temperature);
 Serial.print("°C | Hum: ");
 Serial.print(humidity);
 Serial.print("% | Relay: ");
 Serial.println(relayState ? "ON" : "OFF");

 delay(2000);
}
```

IX. FUTURE SCOPE

1) Advanced AI & Machine Learning: Predictive models analyzing weather, soil, crop type to schedule irrigation perfectly, even predicting growth patterns.
2) Precision Agriculture: Moving beyond basic automation to hyper-localized, data-driven resource management.
3) Human-Machine Collaboration (Industry 5.0): AI-powered AgroBots and systems that work alongside farmers for complex tasks.

X. CONCLUSION

The smart agriculture irrigation control system using a soil moisture sensor, water pump, and NodeMCU provides an efficient, automated solution for modern farming practices. By continuously monitoring soil moisture levels, the system ensures that crops receive the optimal amount of water, preventing both under-irrigation and overwatering, which contributes to better crop health and higher yields.

REFERENCES

[1] E. N. Ifeagwu & A. M. O. Obiageli, "Design And Implementation Of Smart IoT Based Plant Irrigation System Using NodeMCU ESP8266 Microcontroller And Blynk Interface Technology", Caritas Journal of Engineering Technology, Vol. 4 No. 2 (2025).
[2] Negm Eldin Mohamed Shawky, "IoT- Smart irrigation with tracking system using NodeMCU ESP8266", International Journal of Scientific Research in Computer Science and Engineering.
[3] Shaik Althaf, Shaik Jakeer Hussain, Lakshmi Srinivas Dendukuri, "Smart Sensors and NodeMCU ESP8266-Based Automated Irrigation System for Effective Water Management in Agriculture", IJRASET, DOI:10.22214/ijraset.2024.64348
[4] Arunkumar C., Mathesh R., Mohamed Saabir M., Mahalingam P., "Smart Irrigation System Using Moisture Level Sensor", IJRASET, May 2023.
[5] Palomi Gawali, Omkar Rode, Ashwin Saundane, Sahil Shaikh, Arman Shaikh, "Developing a Smart Irrigation System Using NodeMCU", IARAS, 2024.